Loway
research

QueueMetrics
call center monitor

Running a small call-center with QueueMetrics and Trixbox

Running a small call-center with QueueMetrics and Trixbox

Version 4.0, Loway

Table of Contents

Preface

Revision history:

- *4.0:* Using HOTDESKING. June 30, 2010.
- *3.0:* New format, joined Inbound and Outbound guides. March 11, 2009.
- *3.1:* Added AsteriskNOW 1.5.0 guide. April 20, 2009.
- *3.2:* New QueueMetrics configuration wizard added in 1.5.2. April 21, 2009.

Chapter 1. Turning TrixBox or AsteriskNOW into a call-center solution

If you are reading this, you probably already know that using TrixBox (TB) or AsteriskNOW 1.5.0 you can install a fully featured Asterisk-based PBX in a matter of minutes. This PBX is very good for most users as it is preconfigured to handle the most common scenarios one can find in a personal environment or in a small to medium sized office.

Still, the typical usage patterns of a small call center are quite different from the ones you are likely to find in a classical PBX, because:

- Agents spend almost their whole working day available to answer the phone (as opposed to standard PBX users who use the phone occasionally while doing other work); handling and answering calls for them has to be made as easy as possible.

- A call center is usually a high-density commercial enterprise; therefore it has to be run and monitored using tools that are able to see immedately how things are going, identify bottlenecks quickly and address them.

In our opinion, running a successful call center is more a state of mind than a given set of telephone gear. What makes a difference is not the number of extensions, queues or agents you have; is a mindset where you consider that the customers calling in are actually very important and you do your best to serve them well within given budget limits.

Maybe you just run a small computer-repair shop and have a couple of lines coming in. What is the cost of having people wait or call multiple times because they cannot talk to anybody within a reasonable time? What will your clients think about you? On the contrary, what will your clients think of your customer service if it always answers on the very first ring? And how do you know if your technicians actually answer the phone when the calls come in or wait five minutes because they are doing other things? These are the questions you should ask. If you follow the guidelines in this document, you'll find an easy way to start answering to questions like these.

Running a call-center, therefore, is not a matter of having multiple PRIs or special hardware. In most cases, you will not even need a separate box from your main PBX running TB. You will only need some software and a bit of configuration to set it up correctly.

Call centers 101: the very basics

Before we start building a small call center, we have to focus a bit on the terminology:

- A *campaign* is a set of calls that belong to the same scope, e.g. your technical support versus commercial support line are different campaigns, though they may be staffed by the same agents.

- An *inbound campaign* is devoted to answering people calling in, while an *outbound campaign* is made up by agents dialing out. Call centers often mix inbound and outbound activities in order to optimize the use of available personnel.

- A *queue* is the physical implementation of an inbound campaign. The queue receives calls and pipes them to the available agents according to a predefined logic (usually, FIFO for the calls and round-robin for the agents). In call center terminology, this functionality is often referred to as the ACD (Automated Call Distribution).

- An *agent* is a person working at a call center. The agent is different from a casual user as an agent logs in and out, in order to tell the system when he is available or not. In this way, the ACD searching logic minimizes agent searching time, as it almost never has to ring up an agent who is not available.

An agent can be working on one or more queues: whenever he is available, all calls coming in to any queue he's working on will be piped to him.

In this tutorial, we will learn how to create both inbound queues and outgoing campaigns and the proper agent setup to handle them successfully.

Prerequisites

To follow this guide, you will need an already-installed, reasonably modern TB or AsteriskNOW instance. It may be your home or office PBX. This tutorial was made for TrixBox 2.8 and its derivatives, but applying it on different versions should be in any case trivial.

You can follow this guide completely while having your PBX running, so there is no need for a prolonged downtime. You will need at least a couple of telephones to test your setup, and a land line you can use.

You should be basically familiar on how to use TB as a basic PBX: creating extensions, connecting to external lines and such things.

Software versions

The following tutorial was created using the following set of software:

- TrixBox CE version 2.8.0.4
- Asterisk version 1.6.0.26-FONCORE-r78
- FreePBX core version 5.5.2.4
- QueueMetrics 1.6.1

or, for AsteriskNOW:

- AsteriskNOW version 1.5.0
- Asterisk version Asterisk 1.4.24
- QueueMetrics 1.5.1

There may be minor differences caused by minor revisions of the software if you have a different version installed.

Tutorial organization

This tutorial is split into two separate parts; one for inbound and the other one for outbound. They can be implemented separately, though system-wide changes are explained only for inbound to avoid duplication.

For each queue/campaign, for both inbound and outbound, we will show how to:

- Define it in TrixBox and QueueMetrics

- Associate agents
- Have agents use their QueueMetrics' agent page
- Run statistical reports and real-time monitoring
- Keep recordings of all calls made and play them back as needed
- Listen to live calls as they are happening

In our scenario, we have a fictious set of queues and agents that work on them. They are defined as:

Queue code	**Campaign name**	**Direction**	**Agents working on it**	**Extensions**
300	Support EN	Inbound	200 and 201	400 and 401
301	Support ES	Inbound	200	400
302	Helpdesk	Inbound	201	401
350	Callback	Outbound	200, 201 and 202	400, 401, 402

Each agent is defined by his/her agent code; as you can see, extension codes are now separate and linked to the physical extension being used. You can have agents log on from different extensions and still track their activities properly.

Chapter 2. Running an inbound call-center

In this example, we will show how to install QueueMetrics on the same server using MySQL storage and configure everything to have a working system. Setting things up should require about 30 minutes.

QueueMetrics is a full-fledged call center monitoring solution - see http://queuemetrics.com . It is an industry-proven, commercial product that is available free of charge to smaller call centers, home users and SOHO's and is used in a large number of call center worldwide, including installations with hundreds of agents online.

Logging in into TrixBox

If you have a SSH client or can access the system console (i.e. the attached keyboard and monitor, if any), log in as user *root* with password *password* (you entered the password during system installation).

If not, you should install the Java SSH client - see *Common problems and solutions* at the end of this tutorial.

Installing QueueMetrics

After logging in as *root*, type the follwing commands:

```
wget -P /etc/yum.repos.d http://yum.loway.ch/loway.repo

yum install queuemetrics
```

The *yum* command will download QueueMetrics and all of its dependences and install it on your system. This may take a while, depending on your internet connection speed. When asked to confirm the installation, type "y" to proceed.

After the installation is done, you have to install the sample MySQL database that will be used to initialize the system by executing the following commands:

```
cd /usr/local/queuemetrics/webapps/queuemetrics-1.6.1/WEB-INF/README

./installDb.sh
```

Note that the exact directory to use will depend on the QueueMetrics version being installed and is displayed on the last page of output that *yum* produces. Follow the on-screen instructions (it is a matter of typing in a couple of passwords as detailed by the installDb utility itself) and the database will be created.

Tip

For your convenience, remember that the default MySQL password for TB is "passw0rd" (yes that's a zero) and the suggested default DB password for QueueMetrics is "javadude".

Tip

For your convenience, remember that the default AsteriskNOW installation has empty default MySQL password and the suggested default DB password for QueueMetrics is "javadude".

Including the [queuemetrics] context in Asterisk

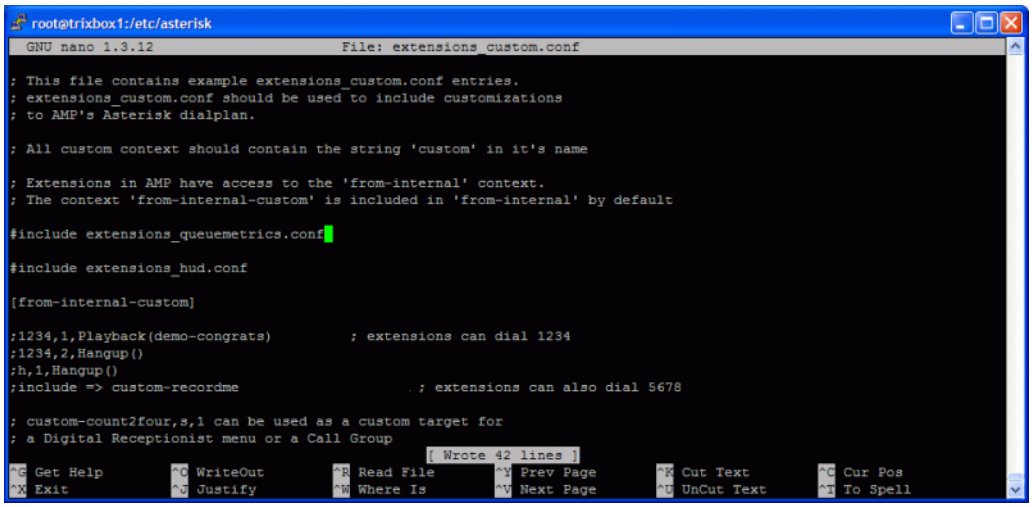
QueueMetrics comes with an user-modifiable Asterisk context that should be included in the main Asterisk context to provide additional functionalities, like e.g. automated agent log-ons and many more.

Run the following commands:

```
cd /usr/local/queuemetrics/webapps/queuemetrics-1.6.1/WEB-INF/

cp mysql-utils/extensions-examples/extensions_queuemetrics.conf /etc/asterisk
```

Then you should edit the file */etc/asterisk/extensions_custom.conf* as shown in the picture by issuing the command:

```
nano /etc/asterisk/extensions_custom.conf
```



Basically you should add a line that says:

```
#include extensions_queuemetrics.conf
```

that will be picked up by Asterisk on the next reload.

Changing QueueMetrics defaults to suit it to our installation

Edit the *configuration.properties* file of QueueMetrics:

```
cd /usr/local/queuemetrics/webapps/queuemetrics-1.6.1/WEB-INF/
nano configuration.properties
```

Look for the following properties across the file and change them accordingly:

```
default.queue_log_file=sql:P001

callfile.dir=tcp:admin:amp111@127.0.0.1
```

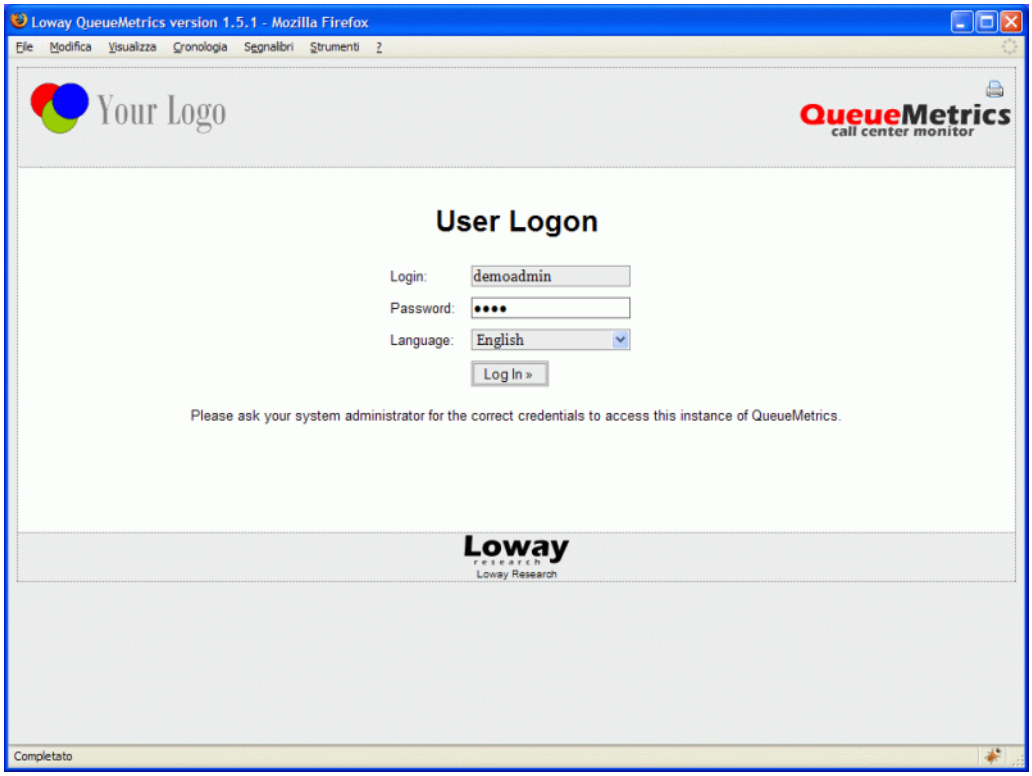
```
default.rewriteLocalChannels=true

callfile.agentlogin.enabled=false

callfile.agentlogoff.enabled=false

default.hotdesking=86400
```

These changes mean:

- use the SQL storage model (see below)
- use the correct credentials to access the Asterisk manager
- rewrite dynamic channel names to the *Agent/XXX* format
- turn off the buttons for old-style agent log-on and log-off.
- use Hotdesking

Testing the installation

To test that everything is okay, you'll have to point your browser to the address *http://myserver:8080/queuemetrics* and you should see a screen like the following one.



If you see this screen, you know that QueueMetrics is working fine. As you'll be curious to check it out, you can login as uder *demoadmin* password *demo*.

Letting QM speak to Asterisk

If you run on Asterisk 1.6, you need to also give the "originate" permission to QueueMetrics:

```
nano /etc/asterisk/manager.conf
```

Add the "originate" permission in both "read" and "write" lines:

```
[admin]
secret = amp111
deny=0.0.0.0/0.0.0.0
permit=127.0.0.1/255.255.255.0
read = system,call,log,verbose,command,agent,user,originate
write = system,call,log,verbose,command,agent,user,originate
```

Save and exit.

Installing the MySQL loader - Qloaderd

In this example, we'll keep a copy of the statistics provided by Asterisk on a database table, this offers a number of advantages:

- Makes general operations faster
- It's lightweight
- Lets you keep a double copy of raw queue data
- Lets you install QueueMetrics on a separate server
- Allows for Hotdesking capabilities

This can be obtained very easily by entering

```
yum install qloaderd
```

Note

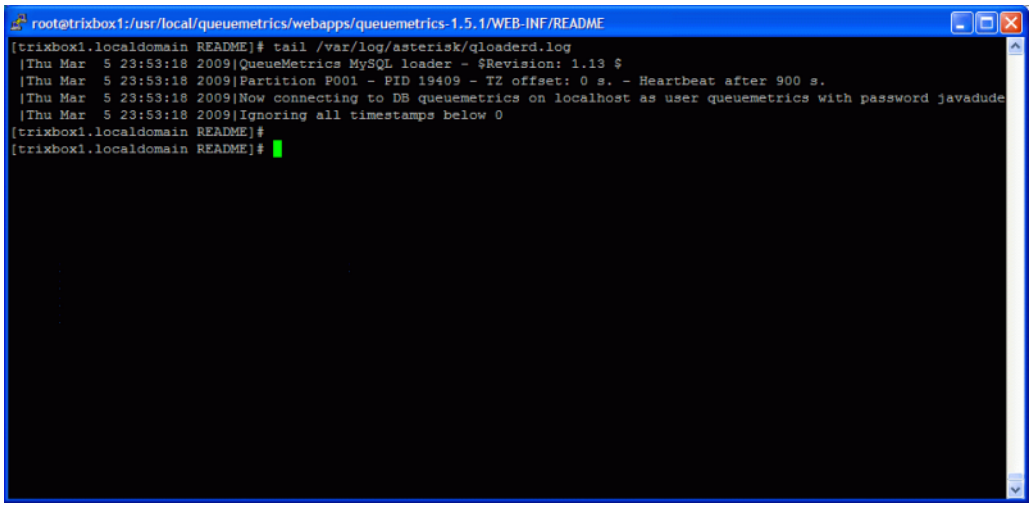
When installing on AsteriskNOW, a warning message will be shown as reported below. Confirm the message with y to continue.

```
warning: rpmts_HdrFromFdno: Header V3 DSA signature: NOKEY, key ID e8562897
Importing GPG key 0xE8562897 "CentOS-5 Key (CentOS 5 Official Signing Key)
<centos-5-key@centos.org>" from /etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-5
Is this ok [y/N]: y
```

After it's done, you can test that the resident loaderd is working by issuing the following command:

```
tail /var/log/asterisk/qloaderd.log
```

If you see a log file that shows no errors, you're done.



After this, do not forget to tun off the log rotation for the */var/log/asterisk/queue_log* file, or the logs enclosing your precious queue data will be deleted periodically.

See *Common problems and solutions* at the end of this tutorial to turn off log rotation.

Installing QueueMetrics on a different server

Installing on a different server can be a good idea if your call center has over 20 or 30 agents and you don't want to slow down the main Asterisk box when running statistics.

It's very easy to do:

- Install QueueMetrics on the new server and install a local copy of the database
- Create a rule on the new QueueMetrics database that allows for connection to MySQL from a client that is located on the Asterisk server.
- Install *qloaderd* on the TrixBox server

When you're done, go to the TrixBox server and edit the file */etc/sysconfig/qloaderd*.

It should look like:

```
PARTITION=P001
QUEUELOG=/var/log/asterisk/queue_log
LOGFILE=/var/log/asterisk/qloaderd.log
LOCKFILE=/var/lock/subsys/qloaderd
PIDFILE=/var/run/qloaderd.pid
MYSQLHOST=localhost
MYSQLDB=queuemetrics
MYSQLUSER=queuemetrics
MYSQLPASS=javadude
```

Edit the variables *MYSQLHOST*, *MYSQLDB*, *MYSQLUSER*, *MYSQLPASS* to point to the new QueueMetrics server.

Then isssue the command:

```
/etc/init.d/qloaderd restart
```

And check the log file to make sure that there are no errors and data is being uploaded correctly to the QM server.

You should also change the *callfile.dir* property in order to point to the Asterisk server and, on the Asterisk server itself, allow for *Asterisk Manager* (AMI) access from the QueueMetrics server.

As a last warning, you should make sure that the Asterisk server and the QueueMetrics server have clocks aligned to a sub-second difference; otherwise the real-time page may act funny, e.g. by specifying negative wait times. In order to avoid this, you should install *ntpd* on both servers.

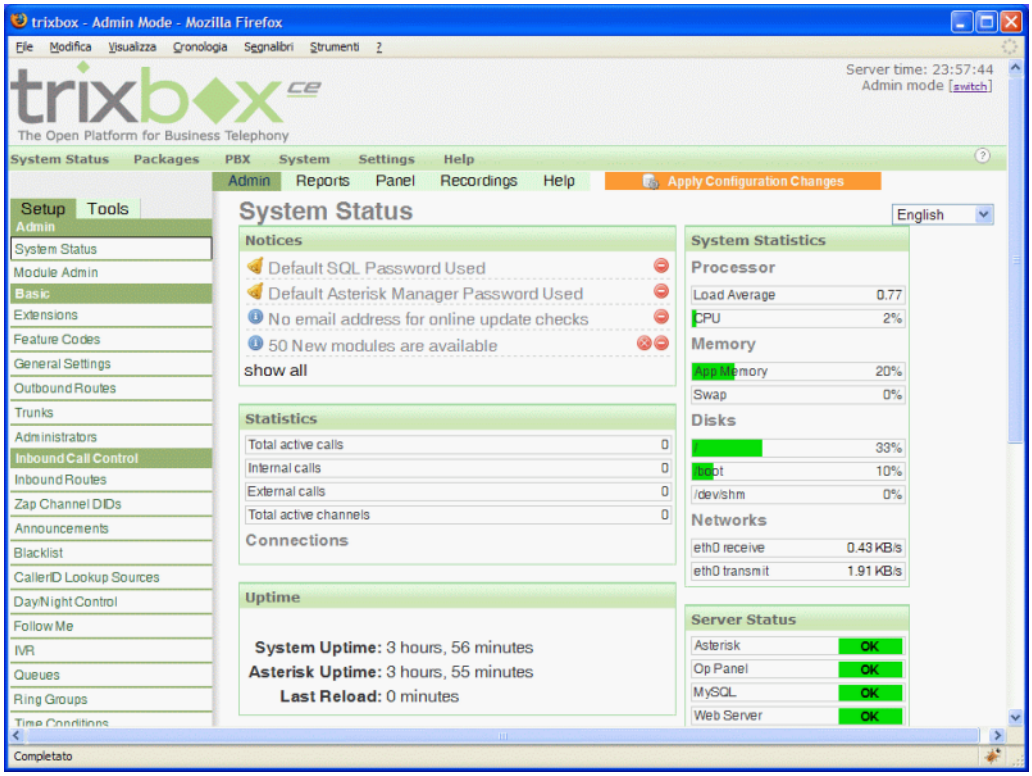
Configuring TrixBox or AsteriskNOW

Point your browser to your TrixBox or AsteriskNOW server; you should see a welcome screen.

When using TrixBox:

- Click on *User mode [switch]*
- Enter user *maint* and password *password* as credentials
- Click on *PBX # PBX settings*

You should see the FreePBX welcome screen.



When using AsteriskNOW:

- Select *FreePBX Administration*
- Enter *admin* and *admin* as credentials

You should see the FreePBX system tatus screen.

Creating inbound queues

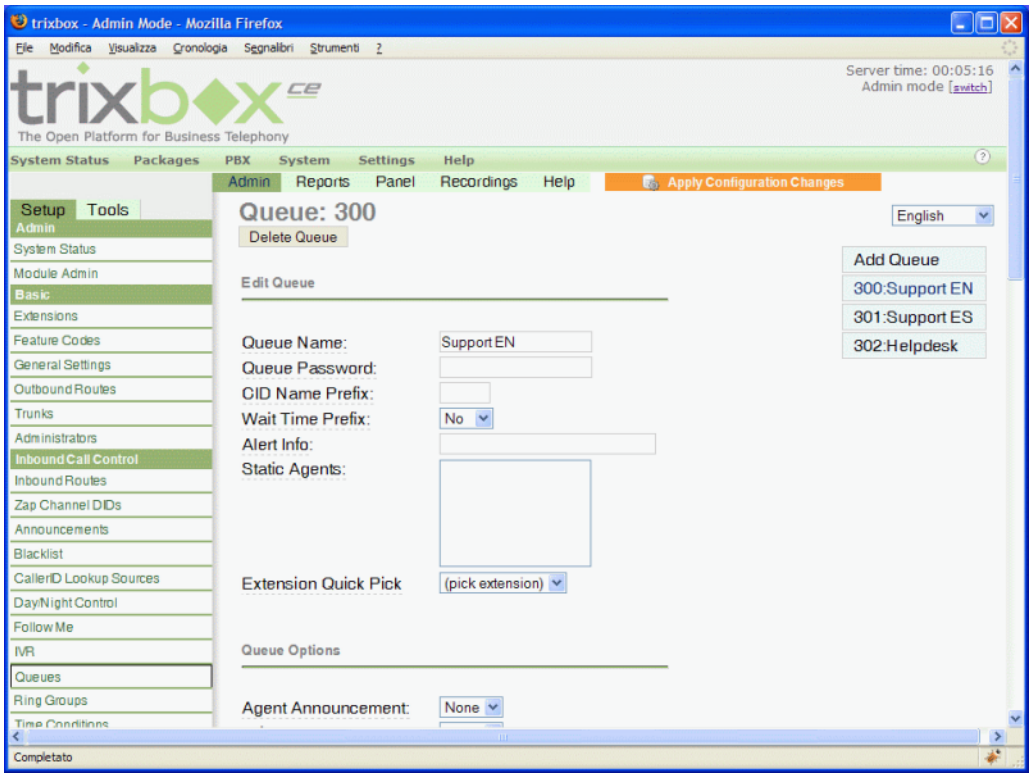
Click on *Queues* and create a new one with the following parameters:

- *Queue number:* 300
- *Queue name:* Support EN
- *Ring strategy:* rrmemory
- *Call recording:* wav49

Leave all other settings blank.

Click on "Apply configuration changes" # "Continue with reload".

You should see a result like the following one.

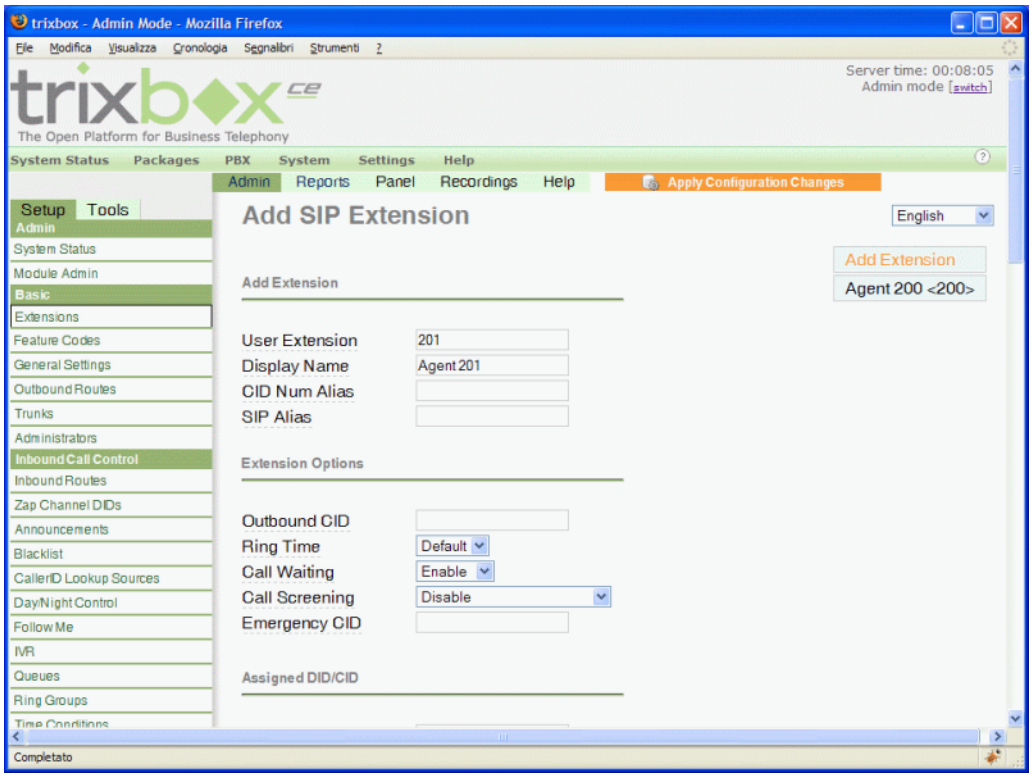


Likewise, create queues 301 and 302 in turn.

Create extensions

If you have not already done so, for the purpouse of this tutorial you should create three SIP extensions and associate them to three physical or software phones.

They should be named 400, 401 and 402.



When you're done, apply configuration changes and try placing some test calls, to make sure that all phones are working.

Configuring QueueMetrics

We now have to configure QueueMetrics to use the newly-defined queues and agents.

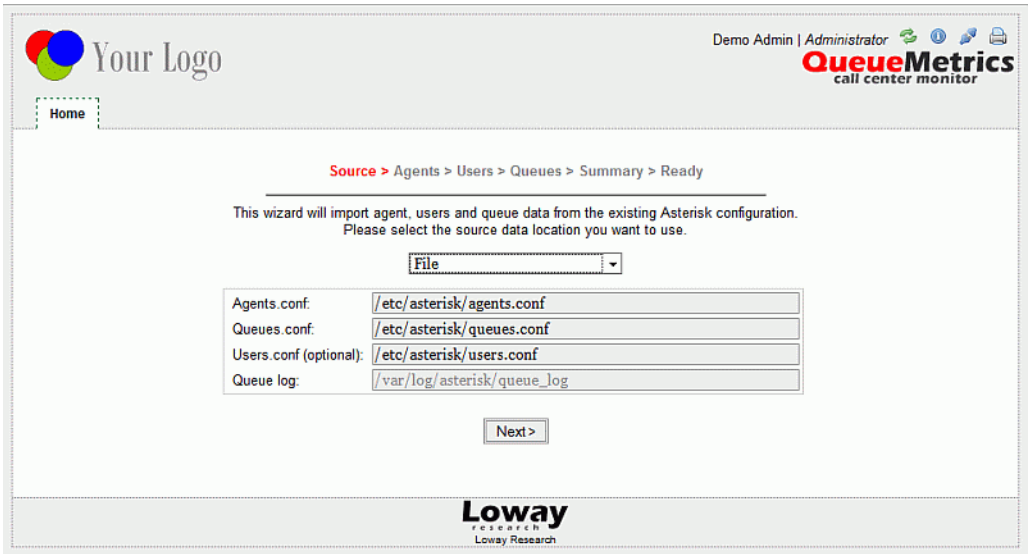
Go to *http://10.10.3.123:8080/queuemetrics* and login as *demoadmin* with password *demo*.

Import queue definitions

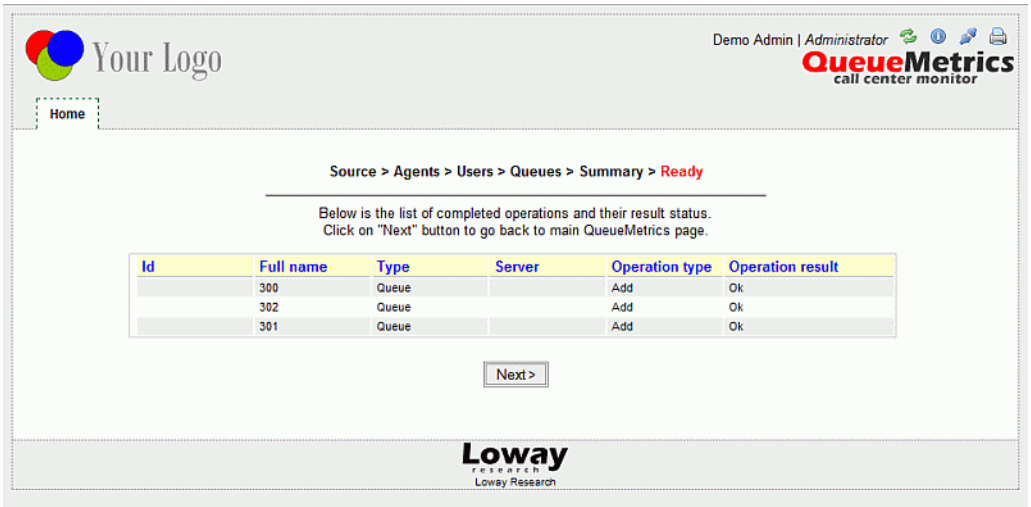
From the home page, click on *Edit QueueMetrics settings # Setup wizard*.

Note

The *Setup wizard* is present only if you're logged with a user holding the CONFIG key. If you are migrating from a QueueMetrics version older than 1.5.2 you need to add this key to your account.



Follow the wizard until complete.



At this point, queues 300, 301 and 302 have been created.

Create agents

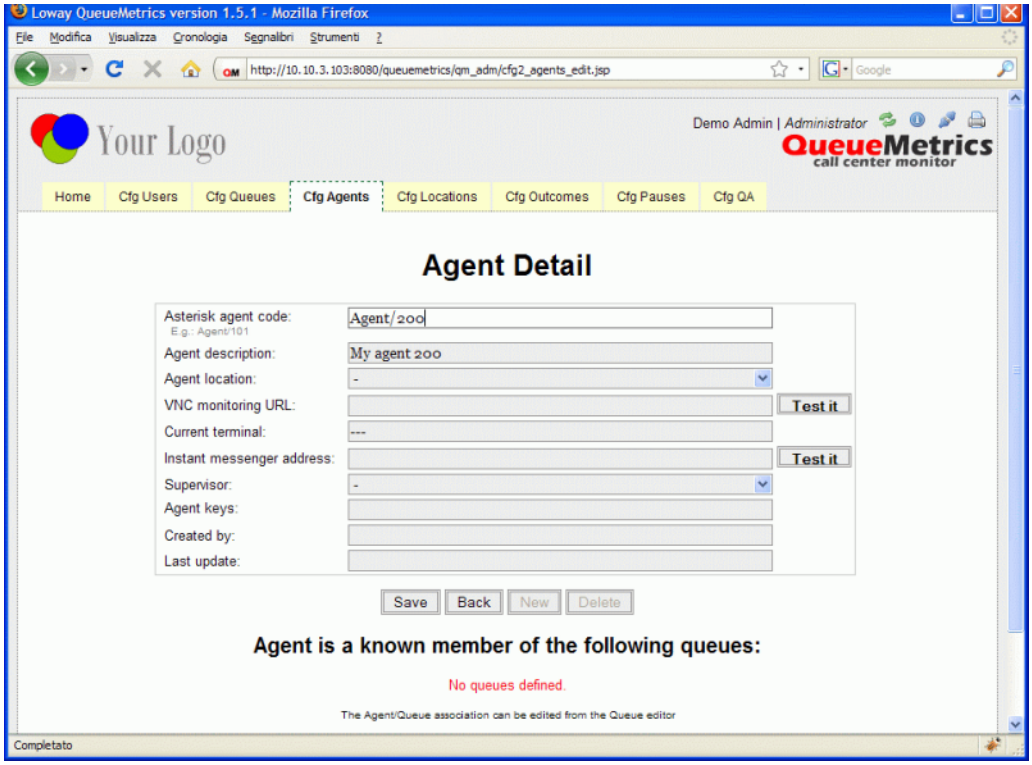
We now have to tell QueueMetrics on which queues our agents will be working.

Edit *QueueMetrics settings # Edit agents*.

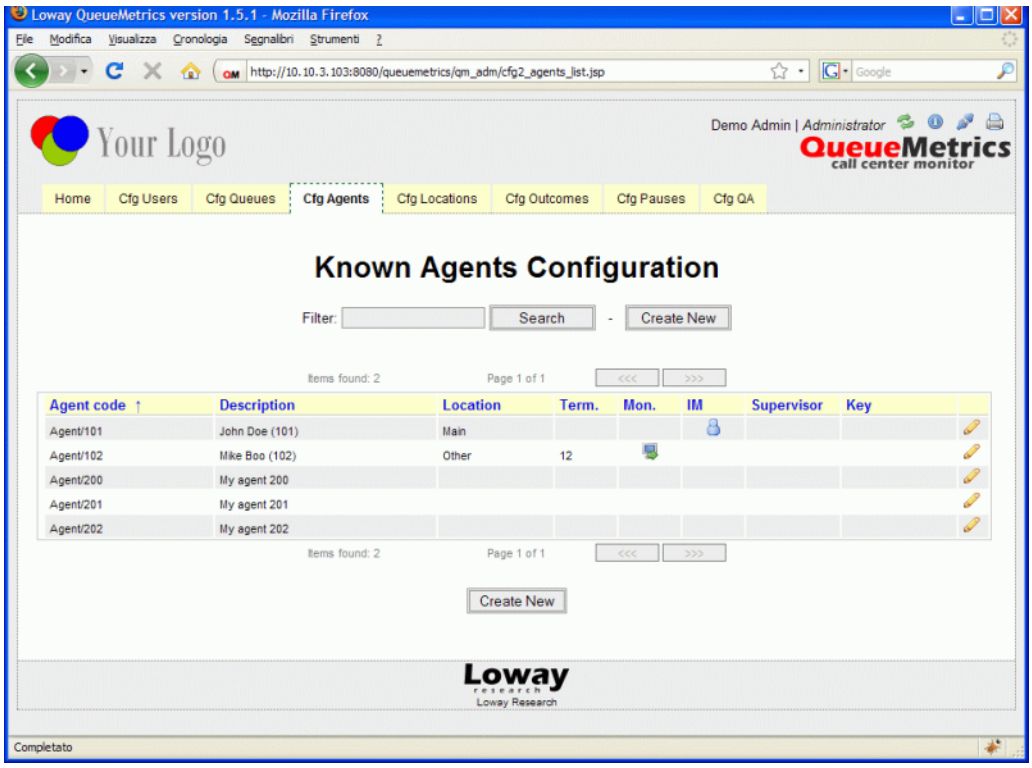
Create a new agent, setting the following parameters:

- *Asterisk agent code:* Agent/200
- *Agent description:* My Agent 200
- *Current terminal:* ---

Leave all other parameters blank. Note that the agent code must be the prefix *Agent/* plus the extension number, without spaces or other characters.

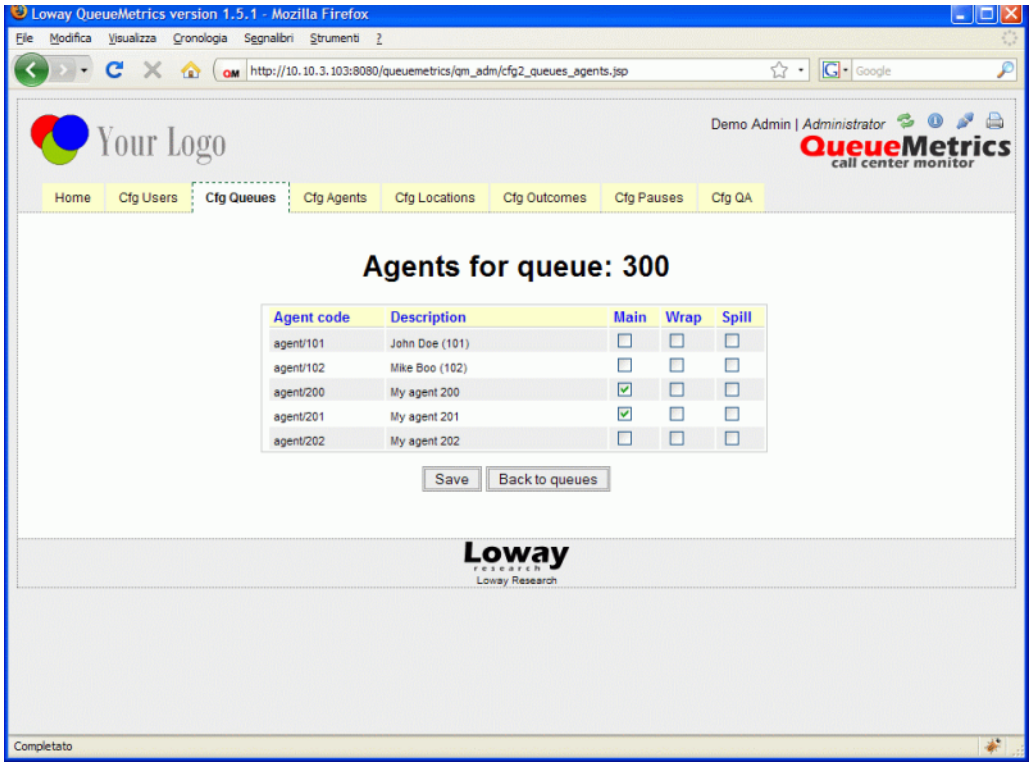


At the end, the agent configuration screen should look like the following one:



Now you should edit the queue-agent association, that is, select which agents can work on which queues.

Just click on *Cfg Queues* # choose queue 300 # *Agents*



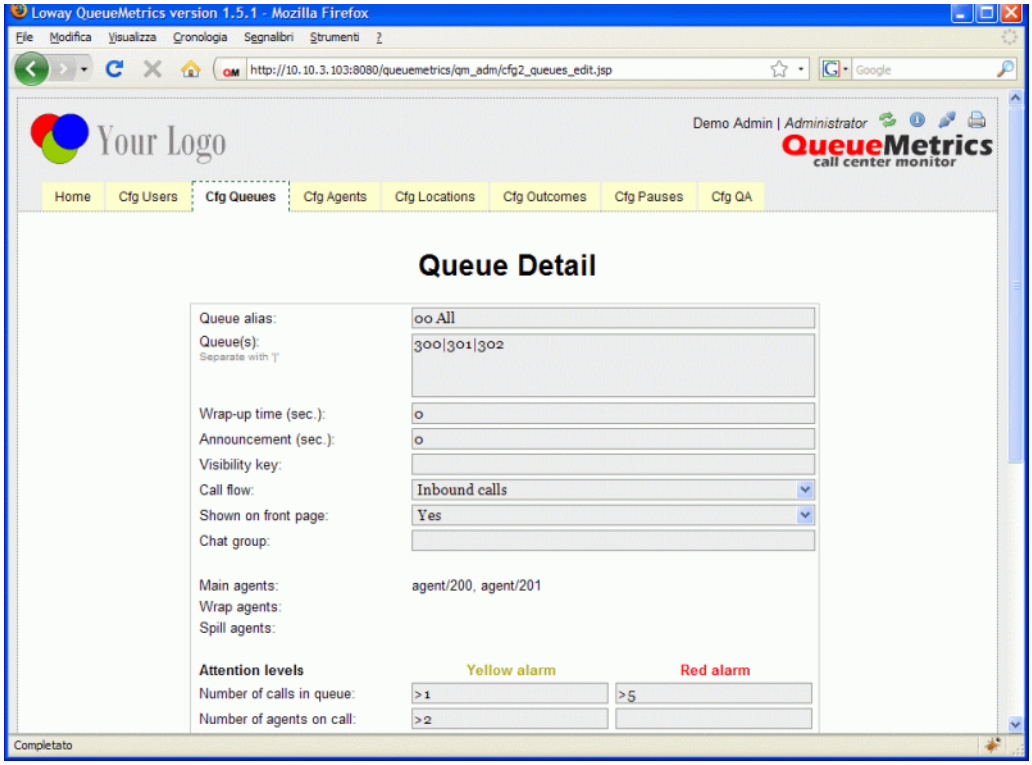
Make sure that the following settings are implemented:

- *Queue 300:* Agent/200 and Agent/201
- *Queue 301:* Agent/200
- *Queue 302:* Agent/201

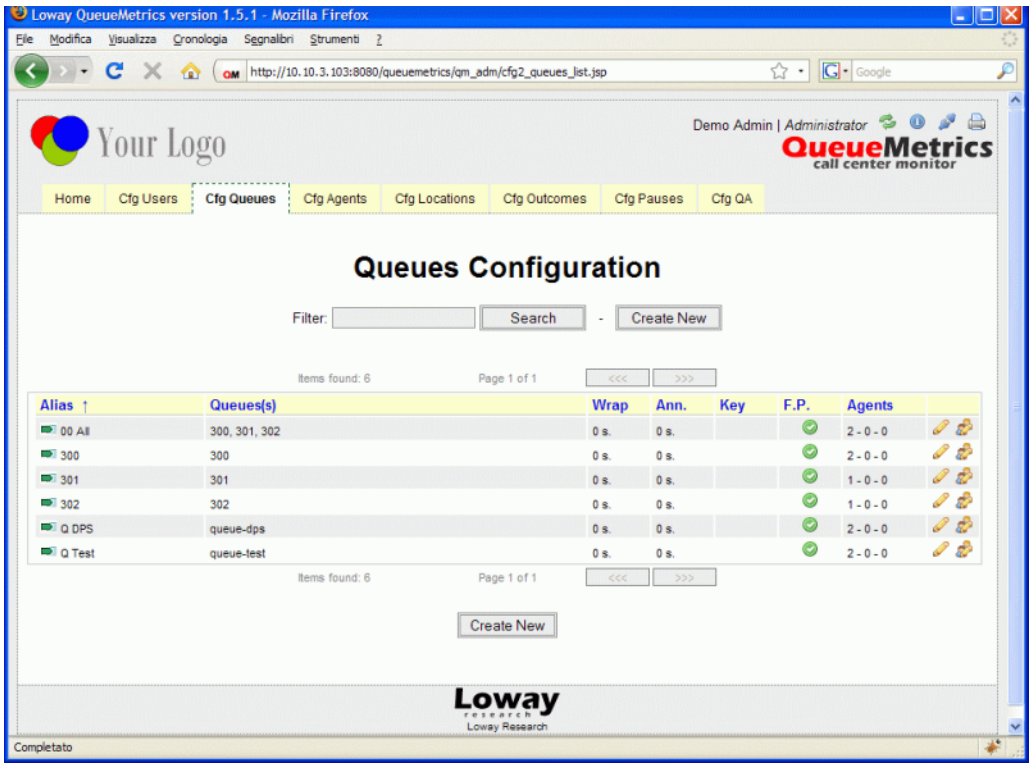
While you are at it, you should also edit the "00 All" queue so that you can see all your inbound activity at a glance.

Set:

- *Queue(s)* to "300|301|302" - this means all of those queues at once
- *Main agents* as Agent/200 and Agent/201



If all OK, the queue configuration page should look like this:



See that *Agents* column shows the number of agents defined as "2-0-0", that is to say 2 agents as Main Level, 0 as Wrap, 0 as Spill.

Creating users

The configuration so far is enough for running reports.

This gets to be a problem if the number of queues and agents is higher than it is in this example - you never can tell if all agents are logged on to their correct queues, and your agents often cannot either.

QueueMetrics helps you in this by offering the so-called *Agent's page*, that is, a specialized page from which agents can log on, log off, go to pause, see calls processed and do other activities as well.

In order to enable this, you have to create a log-on for each agent that matches exactly the agent code you used in the agent definition, so e.g. for extension 200 you would use *Agent/* plus *200*.

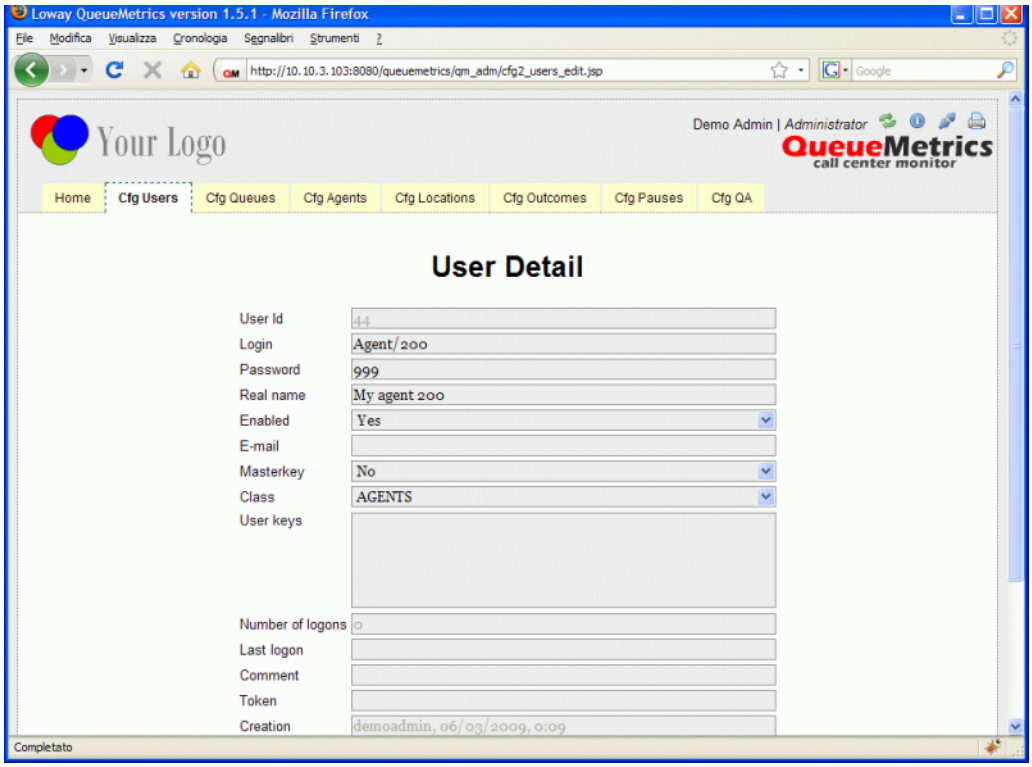
Go to *Home page # Edit QueueMetrics settings # Administer users*.

Create a new user:

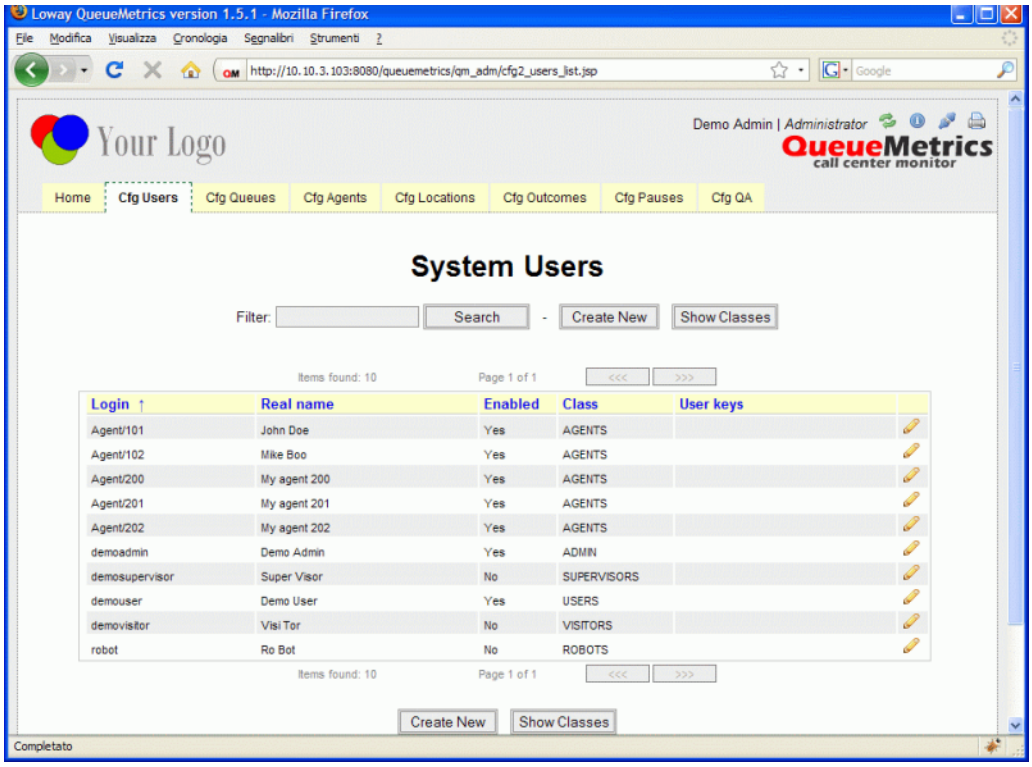
- *Login:* Agent/200
- *Password:* (You choose)
- *Real name:* (The person's name)
- *Enabled:* Yes
- *Class:* AGENTS

Warning

Make sure that the class is set to AGENTS and not e.g. ADMIN, or they will log-on as administrators!



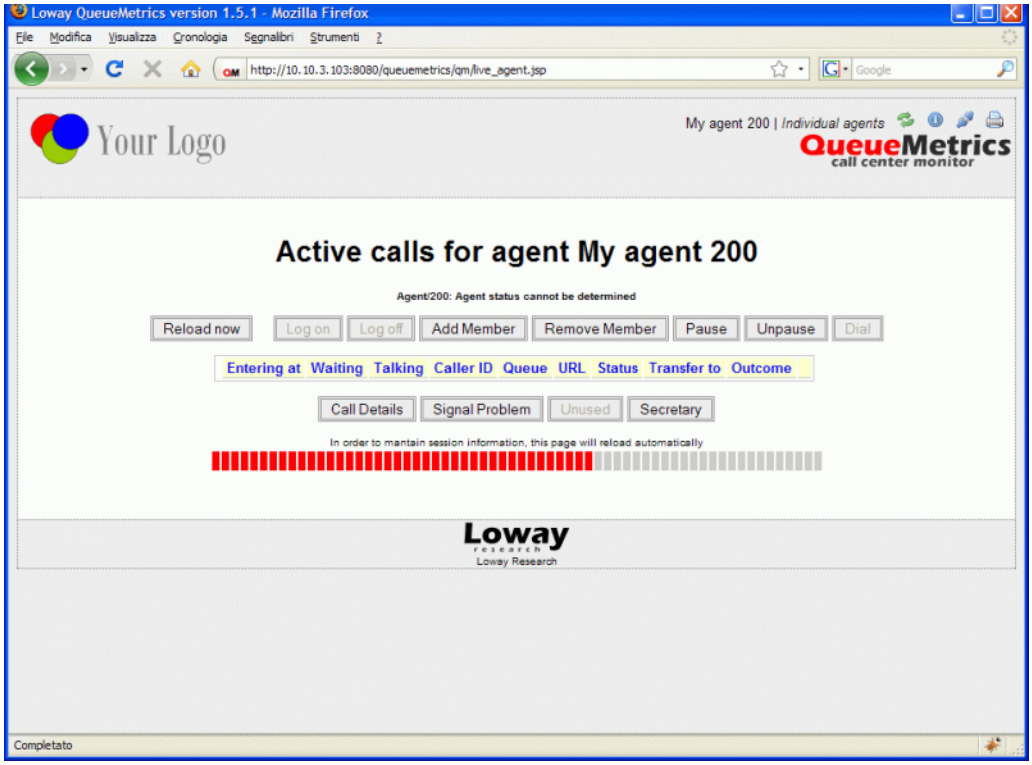
When you are done, create entries for extensions 201 and 202 as well. In the end, the user list should look like the following picture:



Tip

While you are in the user configuration screen, take a second to change the password for user *demoadmin* and the other default users; using default passwords in a production environment is unwise.

To check if your changes have been successful, try logging off and logging in again with the credentials for *Agent/200*; you should see a screen that looks like the following:



From this page:

- To log in to a queue, click on the "Add Member" button, select your queue, enter your local extension and confirm.
- To log off of all queues, click on "Remove Member" and select "All selected" again; you will be logged off in a few seconds.
- An agent can go to pause and mark its pause status as one of the predefined pause codes (e.g. Lunch, Optional break, etc.)
- When taking a call, the agent will be able to mark a "Call status code" for that call (e.g. to mark the call as a Sale)

You can also associate the four bottom buttons to a set of functions that can either be URLs to open or pieces of the Asterisk dialplan to launch.

Running QueueMetrics

You can run QueueMetrics in many different modes:

- As an analytical package, it lets you see who did what in your call-center: how many calls were processed, the response times, agent sessions, etc. It produces over 150 different stats, and it’s fully documented in its user manual that can be downloaded from http://queuemetrics.com/manual_list.jsp (you can also browse it online from the same location)
- As a Quality Assessment package, QueueMetrics lets you gather and analyze statistics on the behaviour of your agents over time.
- As a real-time monitor, QueueMetrics lets you see what is going on in real-time - just select an entry from the queues list and click on "Real-time monitoring"

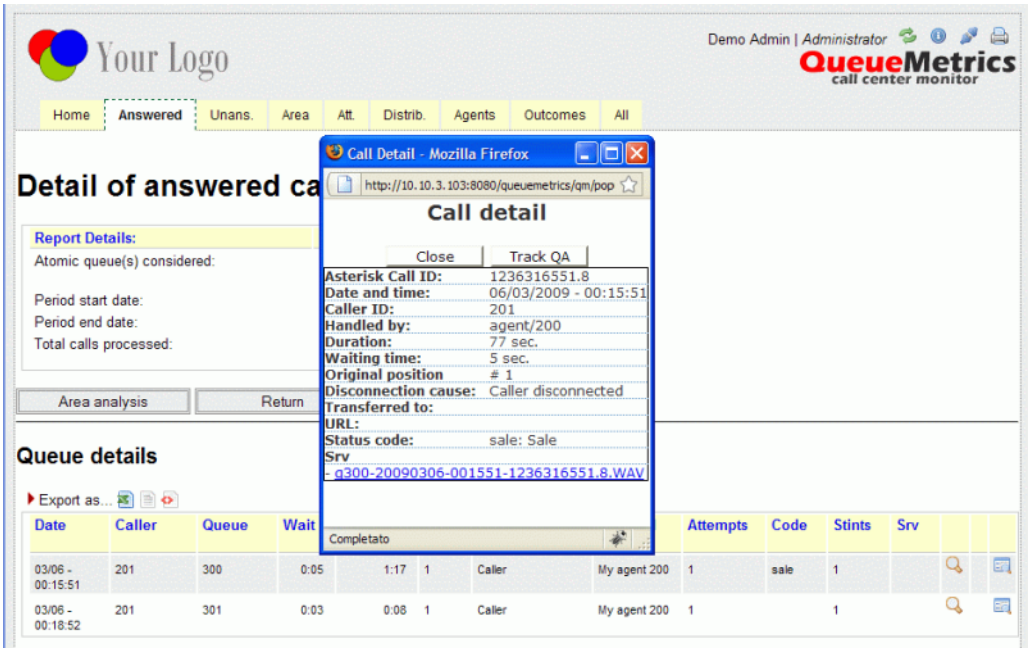
- As a wallboard, it runs a special screen meant to be used with a large screen or video projector; it can be set up so that it's usable from a stand-alone Linux box.
- As an Agent's interface, it will provide your agents with a set of functions that will help them integrate with external CRM apps and perform their tasks more efficiently. It also includes a specialized Firefox app called *AGAW* that acts as a real-time awareness of the general situation for agents.
- As a data source, QueueMetrics will interact with external programs using a standard XML-RPC interface and provide them with high-quality data for further processing.

We suggest that you have a look into the QueueMetrics User Manual to make the most of the wealth of information that QM can provide.

Listening to calls

We configured the system to record all calls in format WAV49 (a derivative of GSM that is natively playable on Windows machines).

Therefore, from QueueMetrics, you simply run a report and then click on "Answered" # navigate to the bottom of the page # "All calls" # see list of all calls found # click on magnifying glass icon # click on the audio file.



Though there is usually one, there can be zero or more audio files linked to a given call.

By listening to the audio recording of a call, you can easily implement a Quality Assessment process to review the quality of each agent's work. Though this topic is not a part of the scope of this giude, it is not to be overlooked and it discussed in detail in the User Manual.

Listening to live inbound calls

It is sometimes nice to be able to listen to live inbound calls as they happen, while the agent is still on line with the caller. QueueMetrics makes such a task trivial.

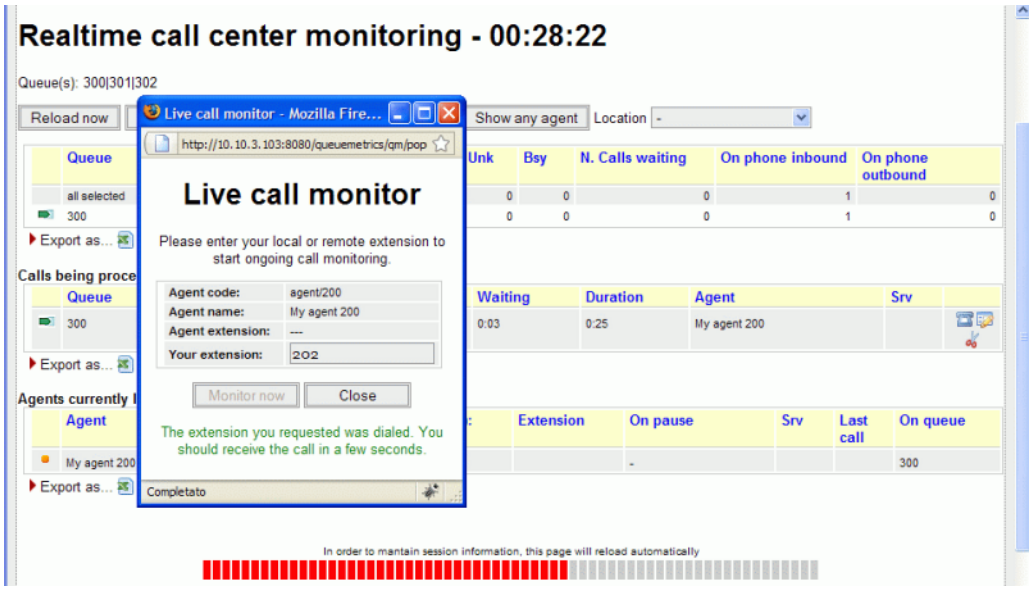
In order to implement this, you must edit the */etc/asterisk/extensions_queuemetrics.conf* and change the following lines as shown here:

```
...
exten => 11,7,ChanSpy(${QM_AGENT_EXT})
...

...
exten => 14,6,ChanSpy(${QM_AGENT_EXT})
...
```

This is needed because we use dynamic agents and not standard agents, so we have to attach to their SIP channels to the actual listening.

After the change, force a reload of the dialplan by making a minor change in FreePBX and applying changes.

After this, go to the Realtime page in QueueMetrics and wait for a call to be available; when it is, click on the small telephone icon and enter your local extension.



Your phone will ring and you will be able to listen to the call in progress right as it's happening.

Chapter 3. Running an outbound call-center

This section of the manual explains how to run outbound call-center activity. We assume that you have already configured your call-center for inbound, as explained in the previous section.

Some things you should know about outbound

In order to run outbound campaigns in your call center, you need to set it up so that agents have a procedure to place outgoing calls.

Why is an "outbound" call different from a normal call?

There are two reasons why "outbound call-center" calls are different from casual calls:

- They are made as a part of one or more ongoing campaigns, and not one-by-one as they come
- You want to be able to distinguish them from "casual" calls made for different purpouses

That's why in our example we use a prefix model where the agent dials a specially formatted number where he specifies the campaign code, e.g. in order to dial number 012345678 for campaign 987, he dials *8987012345678*, as in:

- *8* means outbound campaign dialling, different from your extenal prefix that is usually *0* or *9*
- *987* is the campaign code
- *012345678* is the number to be dialled

As this procedure is complex and error-prone, QueueMetrics offers a web interface that makes it trivial (you just select the campaign from a drop-down list and enter the number to be dialled).

How do I keep track of outbound agents?

As agents that make outbound calls are not, technically speaking, members of a queue (as there is no such thing in Asterisk as an outbound queue), we have a problem trying to understand if they are available or not in the real-time reporting.

As a solution, we suggest to create special queues in Asterisk for outbound presence; these are normal, inbound queues that never get any call, but agents can log-on and log-off from them. This will cause no problem with Asterisk but will make your life easier when monitoring the call-center.

How is outbound activity tracked?

Outbound activity is tracked just like inbound, that is:

- The wait time for a call will be the call set-up time
- The talk time will be the conversation time
- The caller-id will be the *called* number
- The calling person will be show in the *Agent* field

For lost outbound calls, the "agent" field will be displayed.

Can I track inbound and outbound activity at once?

Yes, QueueMetrics lets you track both inbound and outbound activity at once, on a queue-by-queue basis. This is very useful e.g. in the realtime monitoring, so that on a single page you see all of your agents and their current activities, or in the reports to see the total talk time or number of calls related to an agent.

This gets to be misleading when running reports, because the "wait times" and "lost call ratio" of a mixed inbound/outbound queue are meaningless; this is because you do control wait times when receiving inbound calls, but you cannot do the same when calling outside (callees will answer if and when they please). So be careful when doing this.

Configuration changes

Changes to the Asterisk configuration

The first thing you have to do is to specify how yor outgong calls are going to be placed - using an external analog line, or T1, or maybe a direct SIP trunk.

Edit the file */etc/asterisk/queuemetrics_extensions.conf* and look for the following piece of dialplan:

```
[queuedial]
; this piece of dialplan is just a calling hook into
; the [qm-queuedial] context that actually does the
; outbound dialing - replace as needed - just fill in the
; same variables.
exten => _XXX.,1,Set(QDIALER_QUEUE=q-${EXTEN:0:3})
exten => _XXX.,n,Set(QDIALER_NUMBER=${EXTEN:3})
exten => _XXX.,n,Set(QDIALER_AGENT=Agent/${CALLERID(num)})
exten => _XXX.,n,Set(QDIALER_CHANNEL=SIP/${QDIALER_NUMBER})
exten => _XXX.,n,Set(QueueName=${QDIALER_QUEUE})
exten => _XXX.,n,MixMonitor(Q-${QDIALER_QUEUE}-${UNIQUEID}.WAV|b|)
exten => _XXX.,n,Goto(qm-queuedial,s,1)
```

As you can see, here we set a number of variables that let us decide who is the caller and which number he’s trying to reach, and will set up audio recording for the outbound call.

Very likely you’ll have to change at least the line that defines the channel, e.g.

```
exten => _XXX.,n,Set(QDIALER_CHANNEL=Zap/g0/${QDIALER_NUMBER})
```

Will let you call number 012345678 as *Zap/g0/012345678* that is, will dial it on the fist available channel in your Zaptel card.

You may otherwise have a format like *SIP/myprovider/012345678* that will dial numer 012345678 through a SIP trunk called "myprovider".

You should also add the following piece of code to your *extensions_queuemetrics.conf* file (make sure it’s not already present):

```
; extension 28: agent custom dial
exten => 28,1,Answer
exten => 28,2,NoOp( "QM: Agent Custom Dial. ...." )
exten => 28,3,Set(QDIALER_QUEUE=${OUTQUEUE})
exten => 28,4,Set(QDIALER_NUMBER=${EXTTODIAL})
exten => 28,5,Set(QDIALER_AGENT=Agent/${AGENTCODE})
```

```
exten => 28,6,Set(QDIALER_CHANNEL=SIP/${QDIALER_NUMBER})
exten => 28,7,Set(QueueName=${QDIALER_QUEUE})
exten => 28,8,MixMonitor(Q-${QDIALER_QUEUE}-${UNIQUEID}.WAV|b|)
exten => 28,9,Goto(qm-queuedial,s,1)
exten => 28,10,Hangup
```

This is used by the web interface for assisted dialling. Make the same change to the *QDIALER_CHANNEL* variable that you made in the previous context.

As a last change, add the following section to the dialplan (check if it's not already present):

```
; extension 14 makes remote monitoring possible for OUTBOUND CALLS
exten => 14,1,Answer
exten => 14,2,NoOp( "QM_AGENT_CODE: ${QM_AGENT_CODE}" )
exten => 14,3,NoOp( "QM_EXT_MONITOR: ${QM_EXT_MONITOR}" )
exten => 14,4,NoOp( "QM_AGENT_EXT: ${QM_AGENT_EXT}" )
exten => 14,5,NoOp( "QM_LOGIN: ${QM_LOGIN}" )
exten => 14,6,ChanSpy(Local/${QM_AGENT_CODE:6}@from-internal)
exten => 14,7,Hangup
```

This makes it possible to do live listening of outgoing calls.

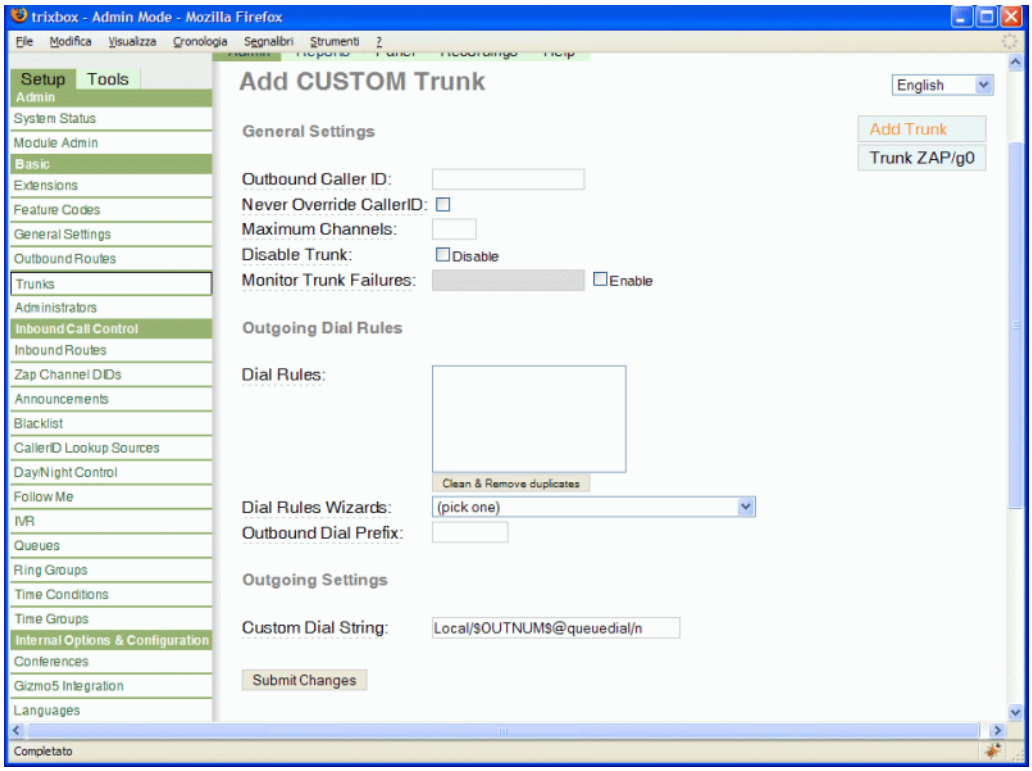
Changes to the Trixbox configuration

Go to *FreePBX # Add custom trunk*

Create a new trunk with the following parameters:

- *Custom dial string:* Local/$OUTNUM$\@queuedial/n

Leave all other fields blank.

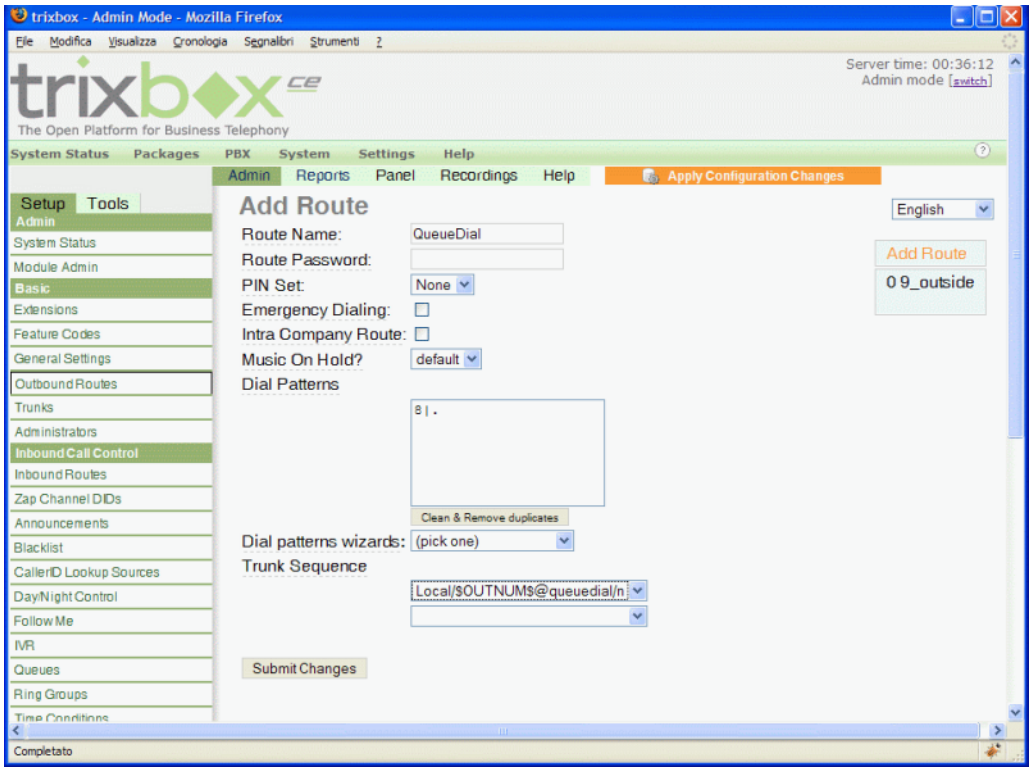


Save the new trunk.

Now you have to create a route to use our trunk.

Click on *Outbound routes # New route* and use the following parameters:

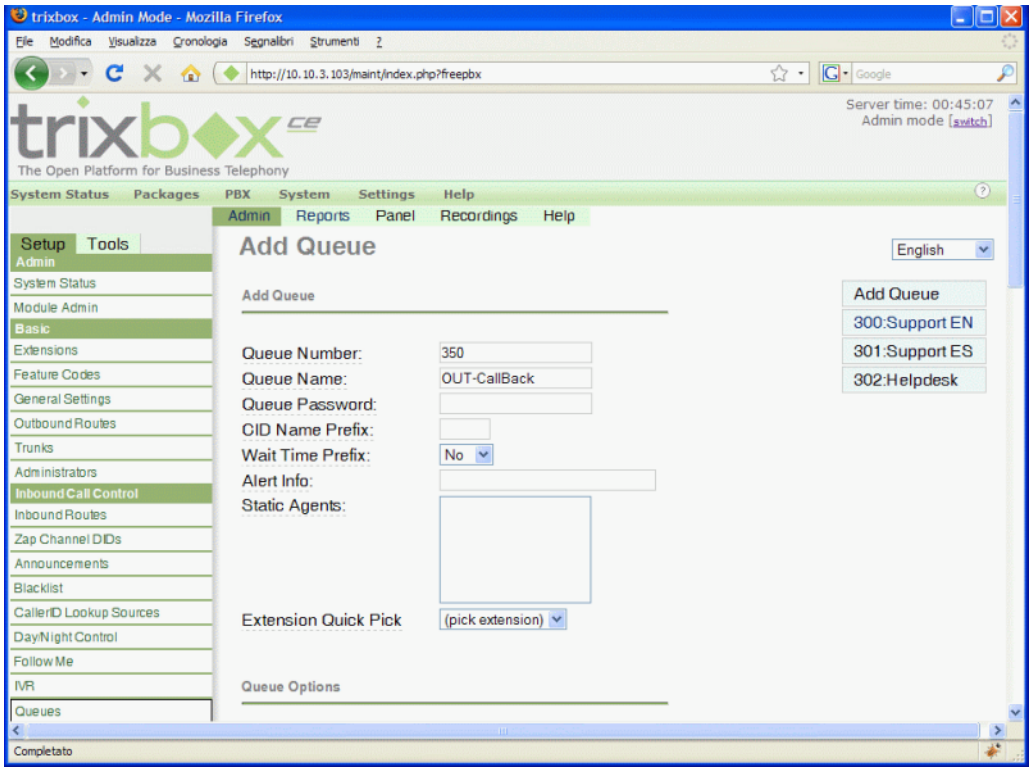
- *Route name:* QueueDial
- *Dial patterns:* 8|. (this means: all dialled extensions that start with 8 match this route)
- *Trunk sequence:* select *Local/$OUTNUM$\@queuedial/n*



Save the new route and reload configuration.

As a last thing, you should create a queue for outbound dialling. Create a new queue like you did for inbound, this time with the following parameters:

- *Queue Number:* 350
- *Queue name:* OUT-Callback



Save and apply changes.

This queue is to be used only as a placeholder to keep track of agent availability status, and will never process any actual call.

Testing the new configuration

At this point try dialing 8123456 from one of the extensions; this tries dialling the PSTN number 456 on campaign 123. Even if you get an error (and this is very much likely, as 456 will not be a valid PSTN number) if you look at the end of the file */var/log/asterisk/queue_log* you will see that a few lines have been appended, e.g.:

```
1179399430|1179399430.13|q-123|NONE|ENTERQUEUE|-|456
1179399430|1179399430.13|q-123|NONE|ABANDON|1|1|0
```

This means everything is in place and working. If now you run a successful call through it, the log will look something like:

```
1179822810|1179822810.22|q-123|NONE|ENTERQUEUE|-|6309886
1179822813|1179822810.22|q-123|Agent/101|CONNECT|3|
1179822823|1179822810.22|q-123|Agent/101|COMPLETEAGENT|3|10
```

Changes to the QueueMetrics configuration

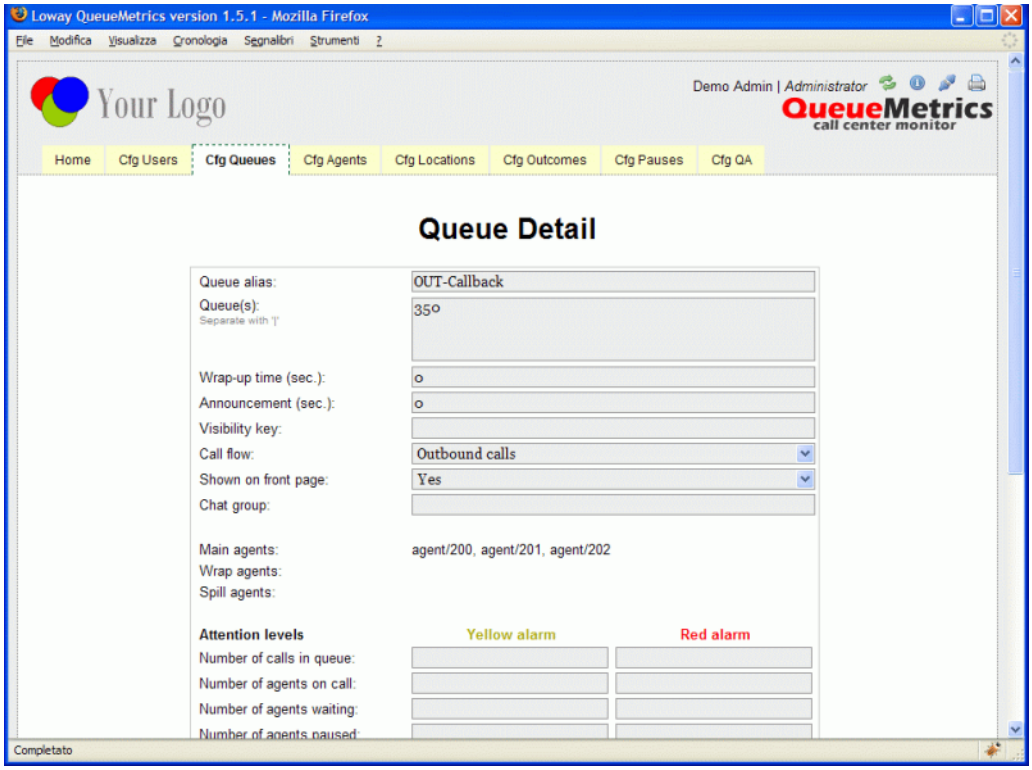
Log in to QueueMetrics and go to *Edit queues*.

Create a new queue with the following parameters:

- *Queue alias:* OUT-Callback

- *Queue(s):* 350
- *Call flow:* Outbound calls
- *Main agents:* Agent/200, Agent/201, Agent/202

Leave other fields blank.



Save and go back to the main page.

Tip

You may want to add this queue to the "00 All" entries, so you can see all activity at a glance; or (better) you can create a new "00 All Inbound" to track all inbound activity separately. See also *Can I track inbound and outbound activity at once?*.

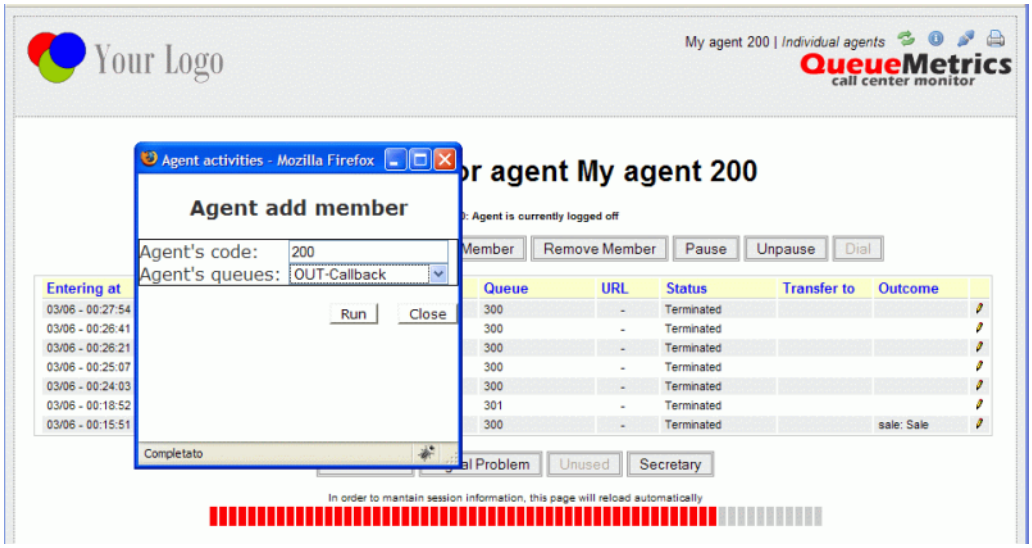
Placing calls

In order for an agent to place a call, you now have two choices:

- Dial the trunk directly, e.g. *8 123 56789*
- Use assisted dialling

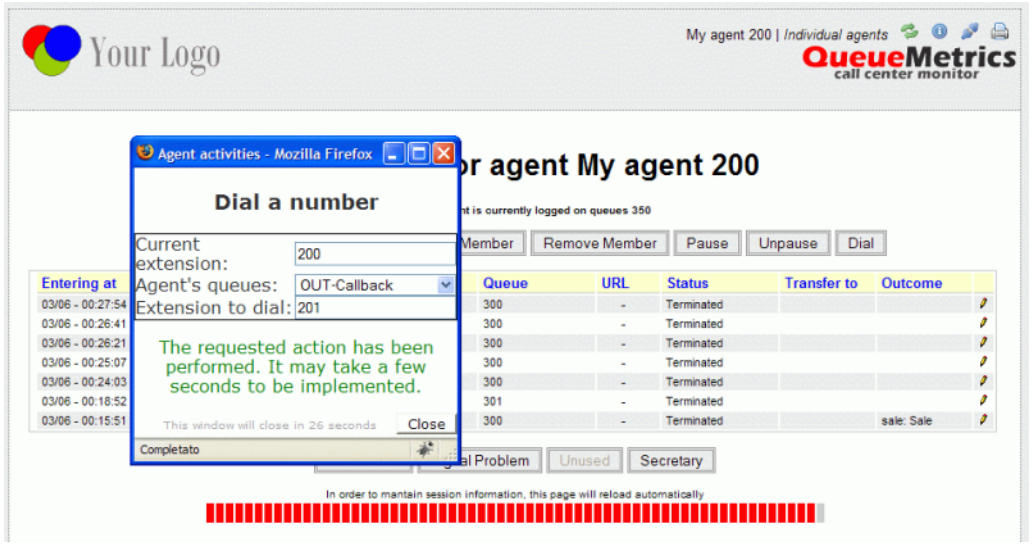
To use assisted dialling, log on as an agent and go to the Agent’s page.

Log on to queue "OUT-Callback" using the "Add member" button.



When you are logged on, click on "Dial".

Enter your current extension, select the campaign and enter the number to be dialled.



The phone will ring; when you pick it up, the outbound number will be dialled.

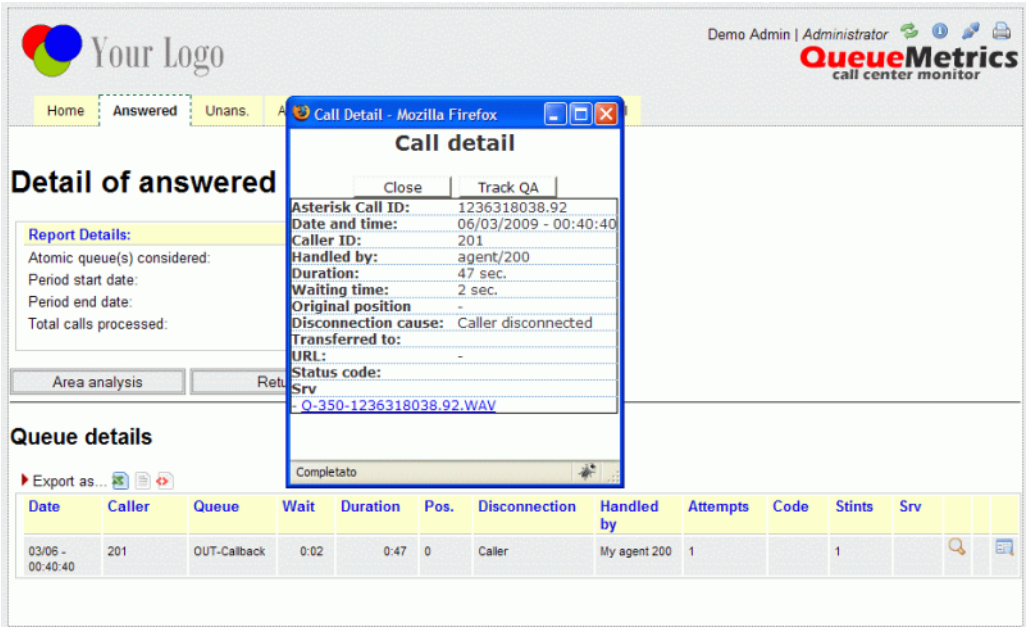
While the call is in progress, you will see call progress as usual:

As always, you can use the Pause and Status keys as you would for an inbound call.

Call listening

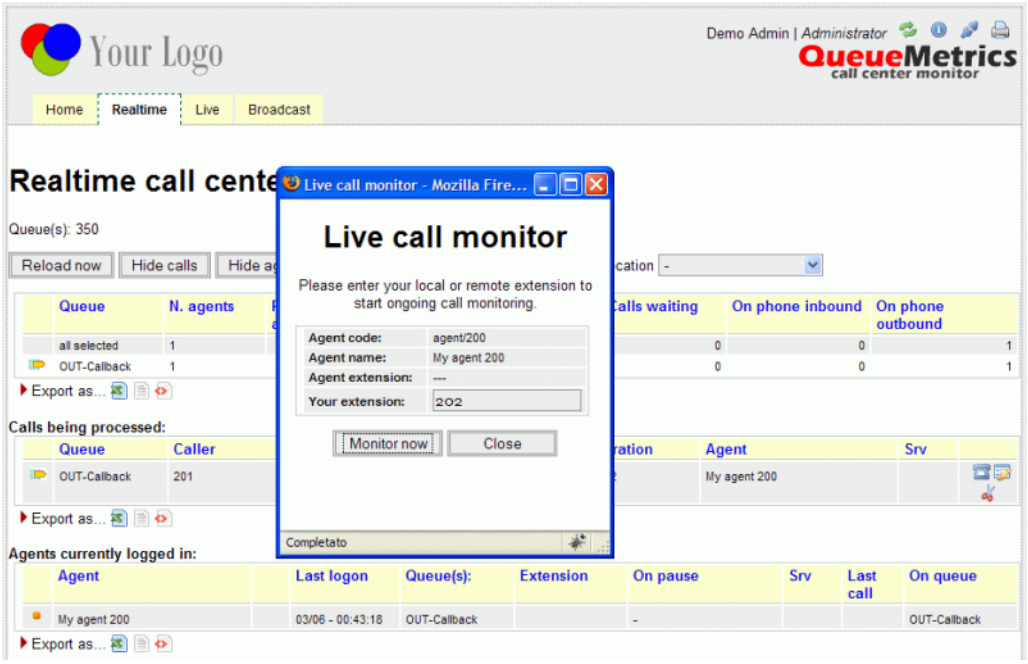
In order to listen to recorded outbound calls, you simply use the same procedure you used for inbound calls:

- Run a report
- Click on "Answered"
- Navigate to the bottom of the page
- Click on "All calls"
- See list of all calls found
- Click on magnifying glass icon
- Click on the audio file.



In order to do live listening, as well, you simply follow the same procedure you did for inbound call, that is:

- Go to the Realtime report
- Wait for a call to be available
- When it is, click on the small telephone icon
- Enter your local extension.



Your phone will ring and you will listen to the call as it’s being made.

Chapter 4. Getting more information

Getting a QueueMetrics temporary licence

We believe that the best way to get a hold of what QueueMetrics is and how useful it is for you is to try it with your own production system. That's why we give a 30-day free evalutation key that you can use freely.

You can get it from: http://queuemetrics.com/sendDemoLicence.jsp

Getting help

If you still are having problems installing or running QueueMetrics on TrixBox, we suggest you check out the following resources:

- The QueueMetrics User Manual is the definitive guide to QueueMetrics. It explains all the features, graphs, reports and configuration option at length. You can read a browsable version at http://queuemetrics.com/manuals/QM_UserManual-chunked/
- The QueueMetrics FAQ at http://queuemetrics.com/faq.jsp are a collection of common solved problems that many people experienced with QueueMetrics. If you are struck by an error message, this is the first place to look at.
- The QueueMetrics forums at http://forum.queuemetrics.com will help you in pinpointing your problems and getting community support. They will also be helpful in seeing what other people are doing with QueueMetrics.
- AstRecipes is a wiki collecting Asterisk "recipes", aimed mostly at call-center users - see http://astrecipes.net
- You may want to contact Loway if your problems are still unsolved - see http://queuemetrics.com/contact.jsp for all relevant contact information.

Common differences between TrixBox and AsteriskNOW

We summarize here the main differences we can find installing AsteriskNOW instead of TrixBox.

- The default MySQL password is empty in AsterisNOW and is *passw0rd* for TrixBox.
- When installing the qloaderd, yum reports a warning message related to a GPG import process; you can confirm the message pressing **y** and the install process will continue.
- The default administration credentials, for AsteriskNOW freePBX GUI are: *admin* as user and *admin* as password.
- The *queue_log* daily or weekly file rotation is disabled in AsteriskNOW.

Common problems and solutions

Avoiding queue_log file rotation

With a standard TrixBox install, the *queue_log* file is rotated daily or weekly along with the other Asterisk logs found in */var/log/asterisk*. The *queue_log* file contains essential information on how the call-center is going that is being used by QueueMetrics to report on the well-being and the actual work being performed by your call-center, and you surely want to keep that data in a safe place for cross-period analysis.

The *queue_log* file is not automatically rotated daily or weekly by the standard AsteriskNOW distribution. You should skip this step.

Disabling log rotation

Disabling log rotation is actually quite easy: go to */etc/logrotate.d* and look for a file named *asterisk*. If you run TrixBox, you'll find something like:

```
/var/log/asterisk/*log {
   missingok
   rotate 5
   weekly
   create 0640 asterisk asterisk
   postrotate
       /usr/sbin/asterisk -rx 'logger reload' > /dev/null 2> /dev/null
   endscript
}
```

There may as well be other sections where other files are rotated. You just delete the section above and the *queue_log* file will not be rotated anymore.

What if my queue_log has already been rotated?

If your *queue_log* has already been rotated, you'll want to join the remaining pieces together. That's very easy:

- First, stop Asterisk and Qloaderd.
- Make a backup of all *queue_log** files
- Rename the existing *queue_log* to *queue_log.now*.
- Execute the following commands (this example is for 5 leftover pieces, you may find a different number on your system):

```
cat queue_log.5 >> queue_log
cat queue_log.4 >> queue_log
cat queue_log.3 >> queue_log
cat queue_log.2 >> queue_log
cat queue_log.1 >> queue_log
cat queue_log.now >> queue_log
```

- Remove all files but *queue_log* itself

- Purge the queue_log table of the Queuemetrics database
- Restart Asterisk and Qloaderd.

This should be it. Now your QueueMetrics will work just fine.

Stopping and starting QueueMetrics

You can stop and restart the QueueMetrics application by issuing the commands

```
/etc/init.d/queuemetrics stop
/etc/init.d/queuemetrics start
/etc/init.d/queuemetrics restart
```

You can also stop and start Qloaderd using the same syntax:

```
/etc/init.d/qloaderd stop
/etc/init.d/qloaderd start
/etc/init.d/qloaderd restart
```

Setting QueueMetrics memory limits

QueueMetrics is a complex application and it is made to be used by multiple parallel users. This means that if you have a large data set and many users running queries on it, it is possible that you start getting "Out of memory" errors.

To fine-tune the amount of memory used by your system, you can edit the file */etc/init.d/queuemetrics* and modify the option:

```
JAVA_OPTS="-Xms128M -Xmx128M"
```

The Xms parameter is the amount of memory that Java uses on startup for its object heap; and the Xmx is its maximum allowed size. For best speed, keep both to the same value unless you have experience in tuning Java memory requirements.

Installing the SSH Java client in Trixbox

If you do not have a SSH client available, you can install one in TrixBox itself by following this procedure:

- Use a web browser and go to *http://myservr/maint* - if asked for login and password, use user *maint* password *password*
- From the PBX drop-down menu, select *PBX settings* # *Module Admin* (on the left-hand menu) # *Check for updates online*.
- Under *System Administration*, click on *Java SSH* and select *Download and Install*, then click the *Process* button at the bottom of the page.
- The system will ask you for confirmation # go ahead and install the module.
- If FreePBX shows an orange label stating that changes must be applied, click on it and apply them.
- From the left-hand menu, now select *Tools*
- You should now find *Java SSH* and click on it

A terminal window will open (it may take a while waiting for the Java client to load).